

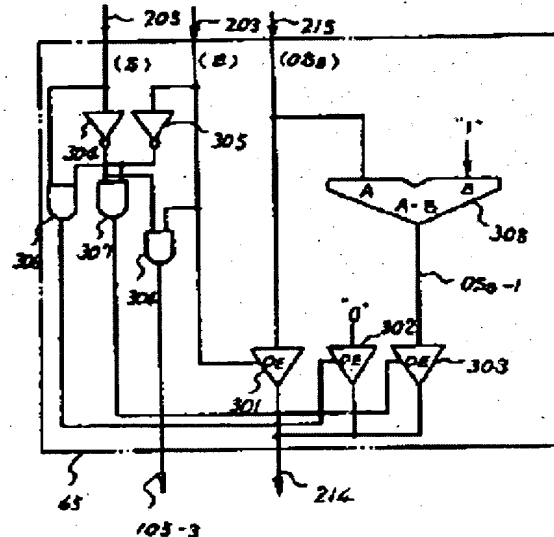
DATA PROCESSING DEVICE FOR ERROR DETECTION OF UNDEFINED LENGTH INSTRUCTION

Patent number: JP58035643
Publication date: 1983-03-02
Inventor: FUKUNAGA YASUSHI; BANDOU TADAAKI; HIRAOKA YOSHINARI; HIRASAWA KOUTAROU; MATSUMOTO HIDEKAZU; IDE TOSHIYUKI; KATOU TAKESHI; NAKANISHI HIROAKI; KAWAKAMI TETSUYA
Applicant: HITACHI LTD;; HITACHI ENG CO LTD
Classification:
 - international: G06F9/30; G06F9/36
 - european: G06F9/30T; G06F9/30T2; G06F9/318; G06F11/08
Application number: JP19810132717 19810826
Priority number(s): JP19810132717 19810826

Report a data error here

Abstract of JP58035643

PURPOSE: To improve the capability of error detection for undefined length instructions, by detecting errors on a basis of contents of the end flag, which is added to an operand designator, and the number of operands.
CONSTITUTION: When an end flag E of an operand inputted to a decode processing end byte number detector 65 is (1), an output gate 301 is opened, and the number of bytes OSB of the operand designator on a bus 215 is outputted to a bus 214. In this case, unless a stop bit S of the operand designator is (1), it means that the number of operand designator is larger than the number of operands, and this is detected as an error. Then, an AND gate 304 is turned on to output a signal 105-3, and the error occurrence is transmitted to an address calculating unit of a CPU. Thus, the capability of error detection for undefined length instructions where the operand designator is used commonly for plural operands to be processed is improved.



Data supplied from the *esp@cenet* database - Worldwide

THIS PAGE BLANK (USPTO)

⑬ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭58—35643

⑤ Int. Cl.³
G 06 F 9/30
9/36
11/08

識別記号

庁内整理番号
6745—5B
6745—5B
7257—5B

⑬ 公開 昭和58年(1983)3月2日
発明の数 1
審査請求 未請求

(全 12 頁)

⑭ 不定長命令のエラー検出を行うデータ処理装置

⑮ 特 願 昭56—132717

⑯ 出 願 昭56(1981)8月26日

⑰ 発 明 者 福永泰

日立市幸町3丁目1番1号株式
会社日立製作所日立研究所内

⑱ 発 明 者 坂東忠秋

日立市幸町3丁目1番1号株式
会社日立製作所日立研究所内

⑲ 発 明 者 平岡良成

日立市幸町3丁目1番1号株式

⑳ 発 明 者 平沢宏太郎

会社日立製作所日立研究所内

日立市幸町3丁目1番1号株式

会社日立製作所日立研究所内

㉑ 発 明 者 松本秀和

日立市幸町3丁目1番1号株式

会社日立製作所日立研究所内

㉒ 出 願 人 株式会社日立製作所

東京都千代田区丸の内1丁目5
番1号

㉓ 代 理 人 弁理士 高橋明夫

最終頁に続く

明 細 書

発明の名称 不定長命令のエラー検出を行うデータ処理装置

特許請求の範囲

1. オペランドのアドレッシングモードを指定するオペランド指定子が処理の種類およびオペランドの数を指定するオペコード部分から独立している命令を扱うデータ処理装置において、該命令は、少なくともオペランド指定子に対応して付加されたオペランド指定子の終了情報を含み、デコード手段は、少なくともオペコードデコード手段と、オペランド指定子デコード手段とオペランド指定子に付加された終了情報の検出を行う終了フラグ検出手段を有し、該オペコードデコード手段が、当該命令の最後のオペランドであることを示す信号を出力したとき、該終了情報検出手段が終了情報のセットを検出しない場合、当該命令をエラーとして検出するようにしたことを特徴とする不定長命令のエラー検出を行うデータ処理装置。

発明の詳細な説明

本発明は、オペランドのアドレッシングモードを指定するオペランド指定子が処理の種類およびオペランド数を指定するオペコード部分から独立している不定長命令のエラーを検出するデータ処理装置に関するものである。

オペランド指定子の長さは、アドレッシングモードに対応して任意に変わり、命令の長さが可変であることからこのような命令を可変長命令と呼ぶこともある。

「不定長命令」と「可変長命令」との間には特別の意味上の差異はなく、「不定長命令」なる用語を「可変長命令」なる用語に置き換えても同一の意味を持つ。しかし、ここでは、便宜上、本願発明が扱う命令を不定長命令、従来の命令を可変長命令と称している。

可変長のオペランド指定子を持つ命令体系として、公知の代表的な2つの例を次に示す。

1つはバローズ社 (Burroughs Corporation) の計算機B1700をCOBOL/RPG 向きのアーキテクチャとした時の命令フォーマットであり、こ

れは、「B1700 COBOL/RPG-S-Language, 1058823-015, Copyright 1973, Burroughs Corporation」に示されている。

今1つの例は、DEC社(Digital Equipment Corporation)の計算機VAX 11/780のアーキテクチャが有する可変長となるオペランド指定子を持つ命令体系であり、これは、「VAX11 Architecture Handbook, Copyright 1979」およびUSP 4236206に示されている。

ここに示した従来の2つの命令体系では、オペランドの形式、アドレッシングモードを指定する部分が、可変長のオペランド指定子で規定され、オペコードから独立であるといった特徴がある。

しかしながら、従来の可変長命令では、処理するオペランドの数と、処理するオペランドのアドレッシングモードを指定するオペランド指定子が1対1に対応づけられており、例えば

$A + B \rightarrow B$

$A + B \rightarrow C$

このような従来例にあつては、命令のエラー検出が困難であつた。

本発明の目的は、複数個の処理するオペランド数に対し、オペランド指定子を共用できる不定長命令のエラー検出能力を向上させたデータ処理装置を提供するにある。

本発明の特徴は、オペランドのアドレッシングモードを指定する各オペランド指定子にそれぞれオペランド指定子の終了情報(以下フラッグと呼ぶ)を付加し、この終了フラッグの内容と、オペコードが指示するオペランドの数によりエラー検出を行うようにしていることである。

つまり、 $A + B \rightarrow B$ という処理では、2つのオペランドを処理し、2つ目のオペランドに対しては、これを2度使用するというのを、オペコードで特別に規定する必要があるからである。

もし、 $A + B \rightarrow B$ という処理で、オペランド数を3とし、オペランド指定子を3つ用意すれば、 $A + B \rightarrow B$ と、 $A + B \rightarrow C$ の区別を意識する必要はないが、 $A + B \rightarrow B$ の処理では、全く同一のオ

ペランド指定子を2つ設ける必要があり、これは、オペランド指定子そのものが、複数バイト(一般に長いものは7バイト)をとる時、メモリの実装効率を著しく低下し、好ましくない。

具体的には、後述するが、オペコードデコード手段がオペランドの最終であることを示す信号を出力した時に、終了フラッグのセットが検出されない場合当該命令をエラーとして検出するようにしていることである。

以下、まず、本願発明の対象である不定長命令を扱うデータ処理装置を図面を参照して詳細に説明する。

第1図は本発明が適用されるデータ処理システムの基本的な概念図である。

メモリ装置1と、複数の中央処理装置2が共通バス3で接続されており、共通バス3を介して相互の間の情報交換が可能となつている。

メモリ装置1は、命令および命令が扱うオペランドを格納するメモリ部11と、この命令およびオペランドの読み出し、書き込みを制御するメモリ制御部12で構成され、メモリ部11とメモリ制御部12はメモリバス13で接続されている。

ペランド指定子を2つ設ける必要があり、これは、オペランド指定子そのものが、複数バイト(一般に長いものは7バイト)をとる時、メモリの実装効率を著しく低下し、好ましくない。

そもそも、 $A + B \rightarrow B$ と、 $A + B \rightarrow C$ の処理を区別することは、メモリの実装効率を高める為になされたものである。($A + B \rightarrow B$ ではオペランド指定子を2つで済むようにした)

このように、従来の方式では、同一のオペランドを複数回使用する処理に対しては、他の同機能の処理と区別する必要があり、オペコードで指定可能な処理数に制限があつた。

尚 $A + B \rightarrow B$ という例で説明したが、これは $A - B \rightarrow B$ という処理も同様であり、AとBの演算をし、その結果をBに格納するという例の全てにいえることである。一般にこれを $A \textcircled{\text{OP}} B \rightarrow B$ と表現している。

メモリ装置1の動作については、特願昭55-160758号明細書に詳細に記載しているが、本発明の要部とは直接関係ないので、この部分の詳

細説明は省略する。

中央処理装置2は、共通バス3に複数台接続可能で(図示では2台)、それぞれ、メモリ装置1より命令及びオペランドをアクセスして順次命令の処理をしていく。

ここでは高速化のため、一度読み出された命令、オペランドをそれぞれコピーしている命令キャッシュ21(高速バッファメモリ)およびオペランドキャッシュ22(高速バッファメモリ)を有し、また命令のフェッチ、デコード及びオペランドアドレス演算を行うIユニット23と、オペランドのフェッチおよび命令の実行を行うEユニットを有し、それぞれがパイプライン処理を行う例を示している。

このような、命令キャッシュ、オペランドキャッシュの使い方、或いは、IユニットとEユニットがパイプライン処理すること自体は公知である。

さて、第2図(A)は、中央処理装置2が扱う不定長命令のフォーマットを示している。

1つの命令は、1ないし数バイトで構成される

ランドが繰返し、使用される。

同一のオペランドを繰返し使用するには、いろいろな実現方法があるが、最も望ましい方法は、最後のオペランド指定子を繰返し使用することであろう。この点については後で詳細に述べる。

オペコードOPで指定するオペランド数と、オペランド指定子の数が不一致の例を次に説明する。

例えば、オペコードOPが加算処理の場合、そのフアンクションは、

$A + B \rightarrow C$

で、3つのオペランドを必要とするが、オペランド指定子が1つの場合は、

$A + A \rightarrow A$

2つの場合は、

$A + B \rightarrow B$

という処理が、同一のオペコードで可能となる。オペランド指定子の具体的な例を第2図(B)に示している。

ここでは、 $\#1 \sim \#24$ の例を示し、そのフォーマットと、これに対応するオペランドを1対1

オペレーションコード(これは通称、オペコードと呼ばれている)OPと、終了フラッグSを伴った1ないし複数バイトのオペランド指定子OS1, OS2……OSnから構成されている。

オペコードOPでは、その命令の処理内容(処理の種類)、処理に必要なオペランドの数およびオペランドの属性(データ長、リード/ライトの区別、データタイプ:固定小数点/浮動小数点……等)が示される。

オペコードOPの後には、当該オペコードOPで示されるオペランドの数以下のオペランド指定子(OS1, OS2……)が示されていて、1つの命令(正式には命令語:インストラクション・ワード)が構成される。

オペランド指定子は、該当命令で使用されるオペランドの順番に並んでいて、最後のオペランド指定子のみ、終了フラッグSが「1」にセットされている。もし、オペランド指定子の数が、オペコードOPで指定されるオペランド数より少ない場合は、最後のオペランド指定子に対応するオペ

対応で示している。

第2図(B)において、オペランドの()は、()内の値をアドレスとしたメモリの内容であることを示している。

また、フォーマット中、DISPは、変位を、IMはイミディエツト(データ直接)を示し、添字はその大きさをビット数で示している。

更に、R₁はインデックスレジスタ、R₂はジエネラルレジスタを示し、Lはオペランドの大きさをバイト単位で示したものである。

第2図(B)において、そのフォーマットとオペランドの関係は、或程度理解できると思われるが、以下、簡単に説明する。

$\#1$ は、レジスタ直接のアドレス指定で、R₁で示されるジエネラルレジスタそのものが、直接オペランドとなるものである。

$\#2$ 以下は、全て、メモリをオペランドとするもので、そのアドレス計算がオペランドの欄に示された形で行なわれる。

$\#2$ は、間接アドレス指定で、R₂で示される

ジェネラルレジスタの内容がオペランドのアドレスとなるものである。

底3, 5, 7では、R_xで示されるジェネラルレジスタの内容に、DISPで示される値が加算されて、これがオペランドのアドレスになつている。

底4, 6, 8では、底3, 5, 7で求められたアドレスのメモリの内容が、オペランドのアドレスとなるものである。

底9~11は、イミディエツトデータで、IM₈, IM₁₆, IM₃₂の値そのものがオペランドとなつている。

底12~17は、ジェネラルレジスタR_xの代りにプログラムカウンタPCが使用されることが、底3~8とは異なるだけである。PCは、デコードするオペランド指定子の次のアドレスを保持している。

底18~24は、底3~8にさらにインデックスレジスタR_xの値が加算されることが異なり、またインデックスレジスタR_xの値は、オペランドのデータ長L分乗算された値が加算される。

27およびアドレス計算ユニット(AU)28部分がこれに対応し、Eユニット24にはオペランドフェツチユニット(OFU)29、実行ユニット(EU)30が対応している。第1図では、Iユニット23とEユニット24がパイプライン処理を行う旨述べたが、それぞれのユニットは第3図に示すように更に命令フェツチユニットIFU25、デコードユニットDU27、アドレス計算ユニットAU28、オペランドフェツチユニットOFU29、実行ユニット30に分割され、それぞれがパイプライン処理をする例を示している。

しかしながら、本願発明の要旨は、このようなパイプライン処理そのものとは直接関係ないので、パイプライン処理については詳細な説明は省略している。尚、パイプライン処理そのものは周知であるが、USP4,025,771号には、パイプライン高速信号プロセッサが示されている。

ところで、第3図において、命令フェツチユニット25は、命令を先行してフェツチするためのプログラムカウンタ50を有し、命令キャツシュ

これは、データ長にかかわらず、インデックスレジスタR_xの値を、先頭からの変位としてセットできるようにするため必要となる処理である。

つまり、L(データ長を示す)を乗算することにより、インデックスレジスタR_xは、データ長にかかわらず、先頭から何番目のデータであるかを示す値を入れておけばよいことになる。

例えば、インデックスレジスタR_xに「10」が入つていると、これは先頭から10番目のデータで、そのアドレスは、バイトの場合は10を加算(L=1)、ワードの場合は20を加算(L=2)、ロングワード(Long Word)では40を自動的に加算(L=4)、ユーザはデータ長にかかわらず、インデックスレジスタR_xの値をセットできる。

第3図は、第1図の中央処理装置2のより具体的なブロック構成図である。

第1図におけるIユニット23は、第3図において、命令フェツチユニット(IFU)25、アライナ(ALIG)26、デコードユニット(DU)

21より次に実行されるであろう命令を先行して読み出す処理を行なつている。

アドレスライン100により、読み出したいアドレスが命令キャツシュ21に送られ対応する命令4バイト分がデータライン101により命令フェツチユニットIFU25に送出される。

命令キャツシュ21に、対応する命令がなかった場合は、共通バス3を介して、メモリ1から該当の命令を読み出し、この命令は命令キャツシュ21にストアされる。キャツシュの動作は周知であり、例えば、「A Guide to the IBM System/370 Model 168」に示されている。

命令4バイト分が命令フェツチユニット25に送出されると、プログラムカウンタ50はプラス4(+4)され、次の命令の送出要求を命令キャツシュ21に出力する。

この動作は、命令フェツチユニットIFU25内にあるバッファ(図示せず)が満杯になるまで続けられる。

命令フェツチユニットIFU25からは、バス

103を介して、あらかじめ読み出しておいた命令がアライナ (ALIG) 26に送出される。

アライナ26は、デコードユニットDU27からの信号線102に指示されたバイト数だけシフト処理を行い、バス104に該当の命令を送出する。

信号線102の値を適当に操作することで、バス104には第5図に示すように、命令の第1オペランド指定子処理時には、左端にオペコードOPが、続いて第1番目のオペランド指定子が並び、2番目以降のオペランド指定子処理時には、1バイトのダミーを^付おいてオペランド指定子が配置して出力される。上記制後^付は、後で詳しく説明する。

デコードユニット (DU) 27は、アライナ26 (ALIG) より送出されたオペコード及びオペランド指定子をデコードして、アドレス計算ユニットAU28へ下記の情報を送る。

(1) バス105を介してアドレッシングモードを送る。

(4) バス108を介してインデックスレジスタR_xのアドレス、

および

(5) バス116を介してアドレス演算に使用するプログラムカウンタの値を送る。

アドレス計算ユニットAU28は、バス105によつて示されたアドレッシングモードに従い、上記(a)、(e)以外の時は、オペランドのアドレス計算を行い、バス109に計算後のアドレスを送出する。

一方、(a)の場合は、バス107の内容を、そのままバス113に送出し、(e)の場合はバス106の内容をバス109に送出する。

オペランドフェッチユニットOFU29は、上記(a)、(e)以外の時は、送られたアドレスが示されているバス109の内容を、バス110に送出し、オペランドがリード時には、オペランドキャッシュユ22にリード処理を要求する。

リードオペランドが、オペランドキャッシュユ22からバス111に送出されると、オペランド

アドレッシングモードには、先に説明の如く、次の(a)~(h)があり、このうち1つが指定される。

(a): レジスタ直接 …… 宛1

(b): R_x …… 宛2

(c): R_x + DISPタイプ …… 宛3, 5, 7

(d): R_x + DISPインダイレクトタイプ ……

宛4, 6, 8

(e): イミディエイト …… 宛9, 10, 11

(f): PC+DISPタイプ …… 宛12, 14, 16

(g): PC+DISPインダイレクトタイプ ……

宛13, 15, 17

(h): (b)~(d)でインデックス付タイプ ……

宛18~24

尚宛1~宛24は第2図(B)に示すオペランド指定子フォーマットの宛1~24に対応している。

(2) バス106を介して、DISPまたはイミディエイトデータを32ビットで送る。

(3) バス107を介してジェネラルレジスタR_xのアドレスを送る。

フェッチユニットOFU29は、バス112を介して実行ユニットEU30に、読み出されたオペランドを送出し、またオペランドがそろつた旨を連絡する。

オペランドがライト時は、実行ユニットEU30からの書き込みデータが、バス111に出力されるまで、オペランドフェッチユニットOFU29は、アドレスをバス110に送出し続ける。

一方、上記(a)に対しては、オペランドフェッチユニットOFU29は、アドレス計算ユニットAU28より送出されたレジスタアドレス113により、自身が有するジェネラルレジスタ(図示せず)のアクセスを行う。(a)以外と異なるのは、メモリアクセスするか、レジスタをアクセスするか^の差異のみである。

また、(e)に対しては、バス109の内容をそのままバス111に送出し、実行ユニットEU30にオペランドがそろつた旨を連絡する。

また、実行ユニットEU30は、デコードユニットDU27からオペコードバス114を介して

送出されたマイクロプログラムの先頭アドレスを受信し、リード時は、バス112のオペランドを用いて、ライト時は、オペランド(データ)をバス111に出力して順次、命令の処理を行う。

また、命令が分岐命令の場合は、バス115を用いて、新たなプログラムカウンタ値を命令フェッチユニットIFU25のプログラムカウンタ50や、後述するデコードユニットDU27内のDPレジスタ69にセットすると同時に、パイプライン処理で先行的に処理されていたオペランドの各ユニットにおける処理結果をキャンセルさせる。

以上が、1つのオペランド指定子に対する処理の概略で、各ユニット(25~30)は、パイプライン処理で、順次オペランド指定子の処理を並列に処理していく。

次に、本発明の要旨に係るデコードユニット27について、具体例を示し詳細に説明する。

第4図は、第3図に示すデコードユニットDU27の具体的な実施例を示すブロック図である。

トはオペコードレジスタ64にセットされる。

オペコードレジスタ64の出力は、該当命令の実行ユニットEU30のマイクロプログラムの先頭アドレスを求めるオペコードデコードユニット61と、該当命令のオペランドに対する情報を有するオペランド情報ROM63に送られる。

ROM61の出力結果201は先頭アドレスレジスタ62にセットされ、オペコードバス114を介して、第1オペランドがオペランドフェッチユニットOFU29から実行ユニットEU30に渡されるのに同期して、EU30に送出される。

ROM63は、例えば第6図に示す構成で、その中には、第6図に示すような情報があらかじめ入力されており、オペコードと第何番目のオペランドの処理であるかの情報をアドレスとして読み出される。

すなわち、オペコードレジスタ64に、第1バイト目がセットされた時には、セレクトSEL81で、バス200側が選ばれるため、オペコードをアドレスとして、その第1オペランドに関する情

DPレジスタ69は、デコードユニットDU27がデコードする命令の先頭を示しており、第1番目のオペランド指定子デコード時は、オペコードのアドレスを、第2番目以下のデコード時は、該当オペランド指定子の先頭-1のアドレスを示している。

上記アドレスは、バス102を介して第3図に示すアライナALIGN26、命令フェッチユニットIFU25に送出されているため、バス104には、第5図に示すように、1バイト目には、第1オペランド指定子の読み出しの場合は、(A)に示すように、オペコードOP、第2オペランド指定子以下の読み出しの場合は、(B)に示すように、ダミーのデータ、第2バイト目には、終了フラグSを含んだオペランド指定子の先頭バイトが、第3バイト目から第7バイト目には、オペランド指定子のその他の情報が出力される。

バス204は、第何番目のオペランドの処理をしているかを示す情報で、本情報が全オペランド処理終了を示している時、バス104の第1パイ

報が読み出される。

読み出された情報としては、

- (1) オペランドの属性、すなわち、リードオペランドであるか、ライトオペランドであるかの情報R/Wや、オペランドのデータ長L(バイト、ワード、ロングワード)を示す情報、
- (2) オペランドの最終であることを示すフラグ、および
- (3) 同一命令の次オペランドの情報が入っているアドレス、

がある。

(1)はバス105-1に出力され、アドレス計算ユニットAU28に出力され、また(2)は、バス203に出力され、デコード処理終了バイト数検出器65に送出される。

また、(2)、(3)の情報は、レジスタ83にラッチされた後、バス204に出力され、次のオペランドを読み出すアドレスとして使用される。

(2)の情報のラッチ情報が、セレクト81の選択

端子Sに入力されるため、(2)の情報が“1”の場合は、オペコードレジスタ64の内容(200)が使用され、“0”の場合は(3)の情報が使用される。

一方、バス104の中で、終了フラッグSを示す信号線205は、デコード処理終了バイト数検出器65に送出される。

また、オペランド指定子の先頭7ビットは、バス206により、オペランド指定子デコーダ66に送られる。7ビットの情報によりオペランド指定子のデコードを行うが、その例を第7図により説明する。

たとえば、第2図(B)の63に示す($R_n + DISP_n$)のオペランド指定子が送られると、第7図(A)のように上位7ビットの中の更に3ビットが010であることを検出して、下記情報が出力できる。

- (1) 2バイト長のオペランド指定子であること、
- (2) バス208の内容をバス106へ出力する場合、DISPの桁合せを行うため3バイトの

- (5) R_n の情報は、オペランド指定子の2バイト目の下位4ビットに存在すること、

の5つである。

以上2つの例につき示したがこれらをまとめてみると次のようになる。

オペランド指定子デコーダ66は、送られてきたオペランド指定子をデコードし、次にあげる情報をそれぞれ出力する。

- (1) バス215へは、オペランド指定子の長さをバイト単位で出力する。例えば、第2図(B)の63のオペランド指定子で、($R_n + DISP_n$)のオペランド指定子が送られてきた時は、“2”を出力する。
- (2) バス211へは、ディスプレイースメント(DISP)/イミディエット(IM)データ用アライナ67に対するシフトバイト数を出力する。
例えば、($R_n + DISP_n$)のオペランド指定子の場合、第7図(A)に示した如く3バイト右シフトを、($R_n + DISP_{32}$)の場合

右シフトが必要であること、

- (3) DISP値として4バイト化を図るため、上位3バイトについてはDISP₃₂の最上位ビットを符号拡張して出力すること、

- (4) $R_n + DISP$ でもつてオペランドのアドレスが計算できること、

- (5) R_n の情報は1バイト目の下位4ビットに存在すること、

の5つである。同様に、第2図(B)の67の($R_n + DISP_{32}$)が送られてくると、第7図(B)に示すように上位7ビットが1110110であることを検出して、下記情報が出力できる。

- (1) 6バイト長のオペランド指定子であること、
- (2) バス208の内容をバス106へ出力する場合、DISPの桁合せを行うため1バイト左シフトが必要であること、
- (3) DISPは32ビット全てが指定されているためそのまま出力しなければならないこと、
- (4) $R_n + DISP$ でもつてオペランドのアドレスが計算できること、

は、第7図(B)に示した如く1バイト左シフトを指示する。

- (3) バス212へは、アライナ67に対するマスクバイトの指示データを出力する。

これは、アライナ67に対し、バス106に出力する4バイトのデータ中、上位2バイト、又は3バイトのマスクを指示することで、1バイトや2バイトのDISP、IM情報の符号拡張による4バイト化を図るためのものである。

例えば、DISP₃₂の時には、第7図(A)に示す如く3バイトシフト、DISP₃₂の時は第7図(B)の如く前に「— R_n —」という余分な1バイトがあるため、1バイト左シフトする。

これは、DISP₃₂の時、上3バイトにはDISP₃₂の符号ビットを拡張して入れておかないと、32ビットの正常なアドレス計算ができないからである。(バス212はその指定のためのもの)

- (4) バス105-2へはアドレッシングモードを出力し、これによつてアドレス計算ユニットAU28の動作モードを指示する。

アドレッシングモードについては、第3図のアドレス計算ユニットAU28の説明に関連して、(a)~(h)の8つのモードがあることを既に説明した。

- (5) バス216へはジェネラルレジスタR_nの存在する位置が、1バイト目か2バイト目かを示す情報を出力する。

(R_n+DISP₁)時は1バイト目、(R_n+DISP₂)時は2バイト目が指示される。

一方、バス108へは、オペランド指定子の中のインデックスレジスタR_xの部分の出力する。

また、セクタ68は、信号216で指定されたジェネラルレジスタR_nの存在する位置(1バイト目か2バイト目かの信号)によつて、R_nに対応する部分(バス207の内容或いはバス210の内容)をバス107へ出力する。

アライナ57は、前述したように、オペランド

E=1であれば、

$$DPINC_n = OS_n$$

これ以外で、S=0であれば、

$$DPINC_n = OS_n - 1$$

また、S=1であれば

$$DPINC_n = 0$$

すなわち、

- (1) オペランドの終了フラッグEが“1”の場合は、該当命令の処理は終了したため、次の命令の先頭をさすように、DPレジスタ69がオペランド指定子のバイト数分(OS_n)加算されるよう信号線214に出力される。
- (2) (1)でなく、また終了フラッグSがセットされていない場合は、次のオペランド指定子が1バイトのダミーを先頭バイトにおいて、バス104に出力させるため、(オペランド指定子のバイト数:OS_n)-1の値が加算されるよう信号線214に出力される。
- (3) (1)でなく、また終了フラッグSがセットされている場合は、DPレジスタ69が、そのまゝ

指定子の2バイト目から7バイト目までがバス208によつて与えられるため、信号線211で与えられたシフト数だけシフト処理を行い、且つ信号線212で与えられたマスク部に対しては、符号の拡張を行い、バス106に4バイトのデータとして出力する。

これらは、第7図(A)、(B)に示した通りである。デコード処理終了バイト数検出器65について説明する。

この部分は、本発明になる不定長命令(Sビット付加したもの)を扱う場合の要部でもある。

デコード処理終了バイト数検出器65には、前述した如く、オペランド終了フラッグEを示す信号線203、オペランド指定子のストップビットSを示す信号線205及びオペランド指定子のバイト数(OS_n)を示す信号線215の3つの信号線が入力されており、信号線214へは、次のオペランド指定子のアドレスを示すため、DP69の加算値DPINC_nをバイト単位で出力する。この場合のアルゴリズムは次の通りである。

の値をとるよう、“0”が出力される。これによつて、同一のオペランド指定子を、次のオペランドに対する処理にも使用することになり、同一のオペランドが繰返し使用されることになる。

第8図は、デコード処理終了バイト数検出器65における上記アルゴリズムを実現するハードウェア構成を示している。

つまり、E=1の場合は、出力ゲート301が開き、バス214にはバス215の内容OS_nが出力され、E=0の時は、S=0のとき出力ゲート303が開き、OS_n-1を出力し、またS=1のとき出力ゲート302が開き、“0”が出力されるようになつている。

本発明においては、Eが“1”の時、Sビットが“1”でなければ、オペランド数より多いオペランド指定子が有ることを意味し、これをエラーとして検出しようとするものである。第8図ではアンドゲート304がオンとなり信号105-3を出力し、アドレス計算ユニットAU28に対し、

該当エラー発生を連絡するようになっている。

アドレス計算ユニットAU28では、信号105-3を以下のユニット(オペランドフェッチユニットOFU29)に連絡し、最終的に実行ユニットEU30にエラー発生が連絡されるようになっている。

尚第8図において、304, 305はインバータ、306, 307はアンドゲート、308は減算器である。

加算器71は、現在のDPレジスタ69の値と、信号線214の値を加算し、セクタ70を介して、DPレジスタ69にセットすることで次のオペランド指定子のアドレスをバス102に出力することが可能となる。これにより、アライナALIGN26は、次のオペランド指定子を第5図(A), (B)に示すフォーマットでバス104に出力できる。

一方、セクタ70によつて、バス115の内容を選択してDPレジスタ69にセットすることで前述した分岐命令におけるDPレジスタ69の

変更も可能となる。

尚加算器72は、DPレジスタ69の値に、該当オペランド指定子の長さを示す信号線215の値(OS_s)を加え、さらにキャリー入力“1”を加えることで、デコードしているオペランド指定子の次のアドレスをバス116に出力する。

アドレス計算ユニットAU28は、バス116の内容をアドレス計算に使用するプログラムカウンタPCの値として利用する。

このように、本発明によれば、オペランド数とオペランドの個数の合理性チェックを行うことにより、エラー検出率を向上させることができる。

尚上記実施例においては、終了フラッグSはオペランド指定子の最上位ビットに付加されているが、必ずしもこの部分に限定する必要はなく、オペランド指定子のどこかに終了フラッグを設ければよい。

また、同一のオペランドを複数回繰返し、利用するのに、上記実施例では、DPレジスタ69の内容を更新させずに、同一のオペランド指定子を

繰返しデコードすることによりこれを実現しているが、これ以外にもデコードユニットから実行ユニットEU30に特別な信号を送り、先に求められたオペランドを繰返し利用するように指示してもよい。

図面の簡単な説明

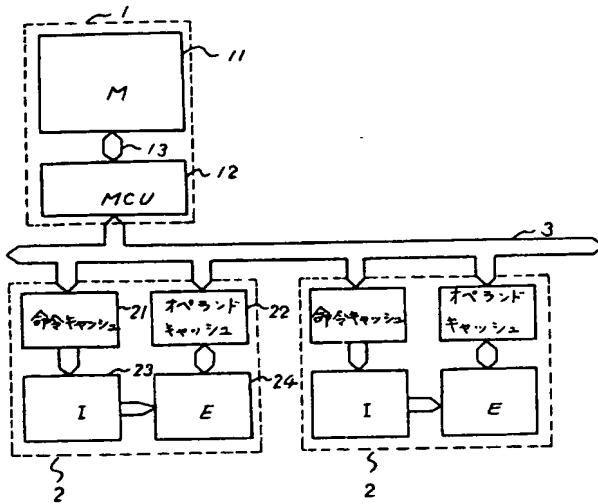
第1図は本発明が適用されるデータ処理システムの基本的な概念図、第2図(A), (B)は本発明に使用される不定長命令のフォーマット及び、オペランド指定子のフォーマットを示す図、第3図は第1図の中央処理装置の具体的な一実施例ブロック構成図、第4図は、本発明の要部である第3図のデコードユニットの具体的な一実施例ブロック図、第5図は第4図の説明に用いられるオペランド指定子のフォーマット、第6図は第4図のオペランド情報ROM63の内容を説明するための説明図、第7図は第4図のオペランド指定子デコーダ66の動作説明に用いる説明図、第8図は、第4図のデコード処理終了バイト数検出器65のハード構成を示す一実施例図である。

27…デコードユニット、61…オペコードデコードユニット、63…オペランド情報ROM、65…デコード処理終了バイト数検出器、66…オペランド指定子デコーダ。

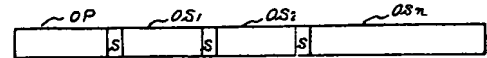
代理人 弁理士 高橋明



第 1 図



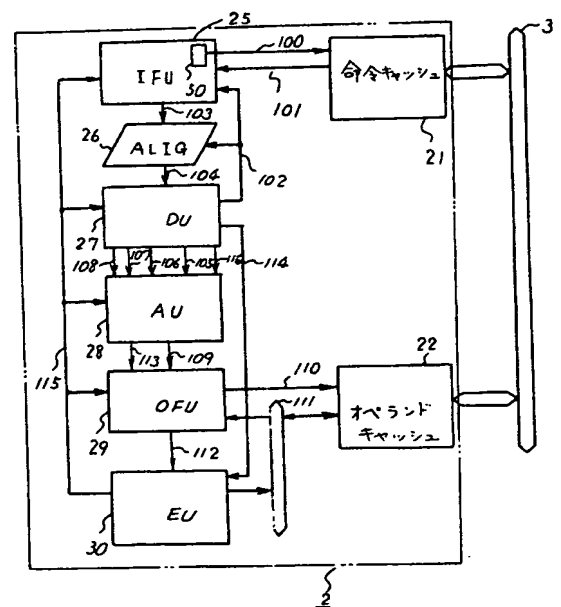
第 2 図
(A)



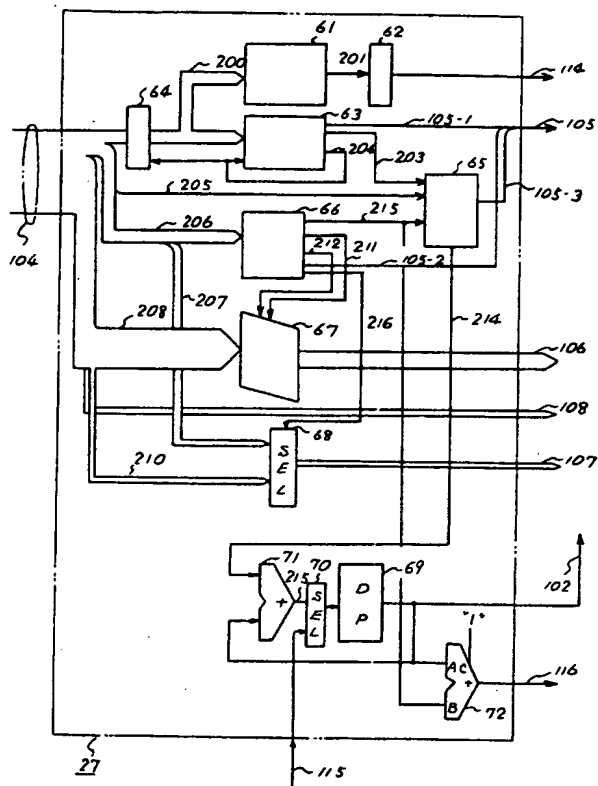
第 2 図
(B)

NO.	フォーマット	オペランド
1	S000 Rn	Rn
2	S001 Rn	(Rn)
3	S010 Rn DISP8	(Rn+DISP8)
4	S011 Rn DISP8	((Rn+DISP8))
5	S100 Rn DISP16	(Rn+DISP16)
6	S101 Rn DISP16	((Rn+DISP16))
7	S111 0110 - Rn DISP32	(Rn+DISP32)
8	S111 0111 - Rn DISP32	((Rn+DISP32))
9	S111 0000 IM8	IM8
10	S111 0001 IM16	IM16
11	S111 0010 IM32	IM32
12	S111 1010 DISP8	(PC+DISP8)
13	S111 1011 DISP8	((PC+DISP8))
14	S111 1100 DISP16	(PC+DISP16)
15	S111 1101 DISP16	((PC+DISP16))
16	S111 1110 DISP32	(PC+DISP32)
17	S111 1111 DISP32	((PC+DISP32))
18	S110 0001 Rx Rn	(Rn+Rx.L)
19	S110 0010 Rx Rn DISP8	(Rn+DISP8+Rx.L)
20	S110 0011 Rx Rn DISP8	((Rn+DISP8)+Rx.L)
21	S110 0100 Rx Rn DISP16	(Rn+DISP16+Rx.L)
22	S110 0101 Rx Rn DISP16	((Rn+DISP16)+Rx.L)
23	S110 0110 Rx Rn DISP32	(Rn+DISP32+Rx.L)
24	S110 0111 Rx Rn DISP32	((Rn+DISP32)+Rx.L)

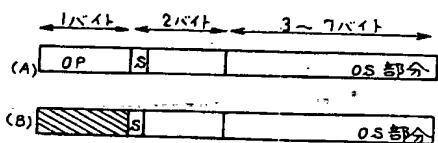
第 3 図



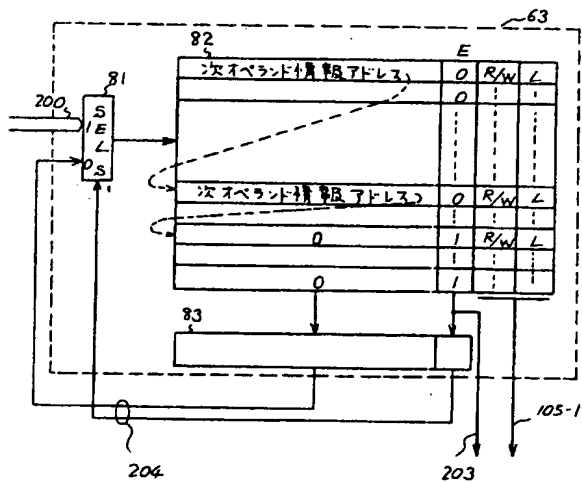
第 4 図



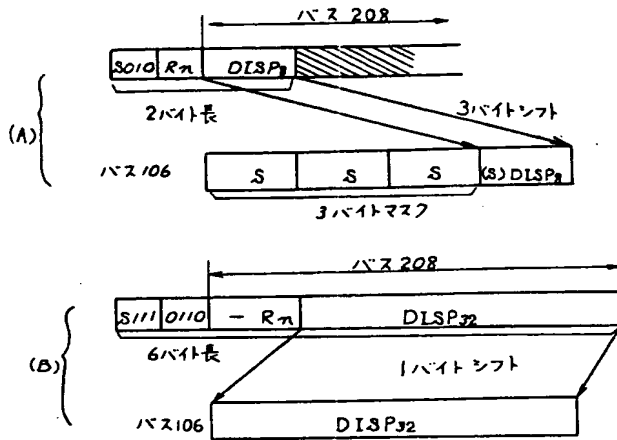
第 5 図



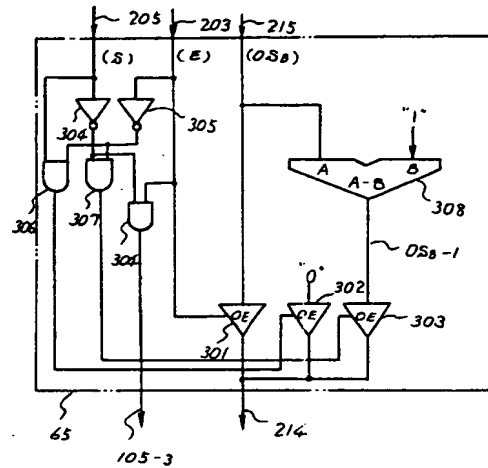
第 6 図



第 7 図



第 8 図



第 1 頁の続き

- ⑦発 明 者 井手寿之
 日立市大みか町 5 丁目 2 番 1 号
 株式会社日立製作所大みか工場
 内
- ⑧発 明 者 加藤猛
 日立市大みか町 5 丁目 2 番 1 号
 株式会社日立製作所大みか工場
 内
- ⑨発 明 者 中西宏明
 日立市大みか町 5 丁目 2 番 1 号
 株式会社日立製作所大みか工場
 内
- ⑩発 明 者 河上哲也
 日立市幸町 3 丁目 2 番 1 号日立
 エンジニアリング株式会社内
- ⑪出 願 人 日立エンジニアリング株式会社
 日立市幸町 3 丁目 2 番 1 号

DOCKET NO: P2001,0304
 SERIAL NO: 10/694,591
 APPLICANT: Hartlieb et al.
 LERNER AND GREENBERG P.A.
 P.O. BOX 2480
 HOLLYWOOD, FLORIDA 33022
 TEL. (954) 325-1100